1. **МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ**
   1. **Постановка задачи**

Автоматизация определения деградации в ответах HTTP-запросов представляет собой задачу классификации, где необходимо определить, является ли ответ на HTTP-запрос нормальным или содержит признаки деградации. Для формализации этой задачи введем следующие обозначения и определения.

* + 1. **Обозначение и определения**

Пространство входных данных: множество всех возможных HTTP – ответов, представленных в виде текстовых строк или структурированных данных.

Пространство выходных данных: , где соответствует нормальному ответу, соответствует ответу с деградацией.

Обучающая выборка: набор размеченных данных

Модель классификации:

Функция ответа x предсказывает метку класса

* + 1. **Цель**

Целью является построение такой модели , которая минимизирует вероятность ошибочной классификации на новых, ранее не встречавшихся данных. Это достигается путем обучения модели на обучающей выборке и последующей валидации на тестовой выборке.

* 1. **Преобразование данных**

Для применения методов машинного обучения и нейронных сетей необходимо преобразовать текстовые данные в числовые представления.

* + 1. **Токенизация**

Токенизация — процесс разбиения текстовых данных на отдельные токены (слова, символы или подслова). Пусть текстовый HTTP – ответ. Токенизация: , где токен, длина последовательности токенов.

* + 1. **Векторизация**

Каждому токену сопоставляется числовой вектор из эмбеддингового пространства. Используется эмбеддинг-функция последовательность векторов:

* 1. **Математическое описание модели нейронной сети**

Для решения задачи классификации применяется нейронная сеть на основе архитектуры трансформера, например, BERT (Bidirectional Encoder Representations from Transformers).

* + 1. **Архитектура модели BERT**

Входные данные: Последовательность эмбеддингов

Трансформер: Использует механизмы внимания для обработки последовательности.

* Механизм самовнимания (Self-Attention)
  + Ключи, запросы и значения:
    - Запрос:
    - Ключ:
    - Значение:

и

* Вычисление внимания:
  + Коэффициенты внимания между токенами
  + Выходной вектор для токена :
* Многоголовое внимание (Multi-Head Attention)
  + Параллельное выполнение нескольких механизмов внимания для захвата различных типов зависимостей.
  + Выходы объединяются и проходят через линейный слой.
    1. **Добавление позиционной информации**

Поскольку трансформеры не учитывают порядок последовательности, вводятся позиционные эмбеддинги , которые добавляются к входным эмбеддингам:

* + 1. **Выход модели и классификация**
* Получение скрытого представления:
  + Используется специальный токен [CLS] в начале последовательности, чье скрытое представление используется для классификации.
* Классификационный слой:
  + Применяется полносвязный слой с функцией активации softmax или sigmoid для получения вероятности принадлежности к классу деградации:
    - Для бинарной классификации с использованием сигмоиды: , где матрица весов входного слоя, b – смещение, сигмоидная функция:
    1. **Функция потерь**

Для задачи бинарной классификации используется бинарная кросс-энтропия: , где y – истинная метка класса, предсказанная моделью вероятность принадлежности к классу деградации.

* + 1. **Обучение модели**
* Оптимизатор: Используется алгоритм AdamW, модификация Adam с коррекцией весового распада потерь по отношению к весам: где параметры модели, скорость обучения, градиент функции потерь.
* Регуляризация:
  + Dropout: Случайное отключение нейронов в процессе обучения для предотвращения переобучения.
  + Весовой распад: Добавление штрафа за большие значения весов
  1. **Метрики оценки качества модели**

Для оценки эффективности модели мы используем следующие метрики.

* + 1. **Матрица ошибок**

Таблица 2.1 — Матрица ошибок

|  |  |  |
| --- | --- | --- |
|  | Предсказано положительное | Предсказано отрицательное |
| Положительное (y=1) | Истинно положительное (TP) | Ложно отрицательное (FN) |
| Отрицательное (y=0) | Ложно положительное (FP) | Истинно отрицательное (TN) |

* + 1. **Точность (Accuracy)**
    2. **Точность (Precision)**
    3. **Полнота (Recall)**
    4. **F1-мера (F1 Score)**
    5. **ROC-кривая и AUC**
* ROC-кривая (Receiver Operating Characteristic) отображает зависимость между долей истинно положительных и ложно положительных результатов при изменении порога классификации.
* AUC (Area Under Curve) — площадь под ROC-кривой, измеряет способность модели различать классы.
  1. **Преобразование задачи в математическую оптимизацию**

Задача обучения модели сводится к минимизации функции потерь на обучающей выборке:

Где параметры модели, N – размер обучающей выборки.

* 1. **Математическое описание процесса обнаружения деградации**

После обучения модель используется для предсказания на новых данных.

* Вход: Новый HTTP – ответ
* Предсказание:
* Решение:
  + Если заданный порог, то считаем, что обнаружена деградация.
  + Иначе считаем ответ нормальным.
  1. **Выводы**

Представленная математическая модель формализует задачу автоматизации определения деградации в ответах HTTP-запросов как задачу бинарной классификации. Использование модели нейронной сети на основе архитектуры трансформера, такой как BERT, позволяет эффективно обрабатывать текстовые данные и выявлять сложные зависимости, характерные для деградированных ответов. Модель обучается путем минимизации функции потерь на обучающей выборке, а ее эффективность оценивается с помощью стандартных метрик классификации.

Таблица 2.2 — Обозначения, используемые в математической модели

|  |  |
| --- | --- |
| **Обозначение** | **Описание** |
|  | Пространство входных данных (HTTP-ответы) |
|  | Пространство выходных данных |
|  | Размер обучающей выборки |
|  | i-й HTTP-ответ |
|  | Метка класса для |
|  | Модель классификации |
|  | Параметры модели |
|  | Функция потерь |
|  | Предсказанная вероятность деградации |
|  | Порог классификации |
|  | Эмбеддинг i-го токена |
|  | Скрытое представление токена [CLS] |
|  | Сигмоидная функция |
|  | Истинно положительные |
|  | Истинно отрицательные |
|  | Ложно положительные |
|  | Ложно отрицательные |

Важно отметить, что модель может быть расширена для многоклассовой классификации, если требуется различать несколько типов деградаций, так же при обучении модели важно учитывать баланс классов в обучающей выборке. При сильном дисбалансе рекомендуется применять методы балансировки.